Mathematics

HP COMPUTER CURRICULUM

Mathematical System

STUDENT LAB BOOK

HEWLETT PACKARD

HEWLETT hp PACKARD

Hewlett-Packard
Computer Curriculum Series

mathematics
STUDENT LAB BOOK

mathematical systems

by E. A. Herber
Golden High School
Golden, Colorado

edited by Jean Danver and Christine Doerr
Hewlett-Packard

Hewlett-Packard Company
11000 Wolfe Road
Cupertino, California 95014

HP5951-4443

April, 1973

The Hewlett-Packard Computer Curriculum Series represents the published results of a Curriculum Development project sponsored by the Data Systems Division of Hewlett-Packard Company. This project is under the directorship of Jean H. Danver.

This material is designed to be used with any Hewlett-Packard system with the BASIC programming language such as the 9830A Educational BASIC, and the 2000 and 3000 series sytems.

The Hewlett-Packard Company makes no warranty expressed or implied, and assumes no responsibility in connection with the operation of the program material contained herein.

TABLE OF CONTENTS

INTRODUCTION

This Mathematics Student Book was written to enrich your study of mathematics by showing you how to use a computer for modeling and as a problem solving tool. The computer is particularly helpful in quickly performing the repetitious steps of algorithms, thus making mathematical investigations easier and more exciting. You will write computer programs which will help you to understand the major concepts involved in the study of a particular mathematical area. If you become more involved in investigating the laws of mathematics, this book will have achieved its aim.

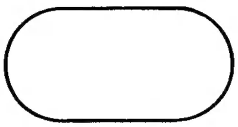
To use the Student Book for Mathematical Systems, you will need the following: First, you should have one year's background in algebra. Second, the book assumes that you already know how to write a program in the BASIC programming language, and that you understand programming techniques for inputting data, performing algebraic operations, designating variables, assigning values to variables, looping, and printing results. If you do not have this background, you will want to study BASIC before attempting this material. Consult the BASIC Manual for the computer you use. Last, in order to complete the exercises in this Student Book, you will need to have access to a computer (about two hours per week for a terminal oriented system). If more time is available, you may be able to experiment further on your own, either to improve your programs or to investigate other areas of mathematics that interest you.

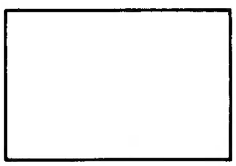
Each section of this book is organized in the same way. First, the mathematical concepts needed to complete the exercises are reviewed. References are listed at the end of each section in case you want to study these concepts in greater detail. Next, each exercise is presented. Finally, an approach is suggested in the Problem Analysis and a flow chart is included to illustrate this approach. The suggested approach was chosen because it brings out the concepts which are being stressed, but the program can sometimes be written more efficiently. Once you have completed the exercise by following the logic in the flow chart, you are encouraged to rewrite the program using more sophisticated programming techniques. You might also want to impose more conditions on the problem to make it more interesting to solve.

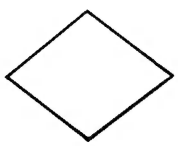
There is no one "right" way to solve a problem by programming. Experiment and learn as you go. You'll find you are learning something new each time, both about your subject matter and about using the computer to solve problems.

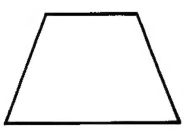
LIST OF SYMBOLS

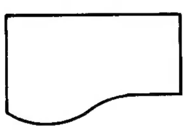
FLOW CHART SYMBOLS

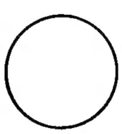

Start or Stop


Defines a process


Represents a decision point


Represents computer input


Represents computer output


Used to connect one part of a flow chart continued at some other place

ALGEBRAIC NOTATION AND EQUIVALENT *BASIC* LANGUAGE SYMBOLS

Algebraic Notation	*BASIC Notation*	*Meaning*
+	+	Addition
−	−	Subtraction
· or X	*	Multiplication
÷ or /	/	Division
$\sqrt{X}$	SQR(X)	Square root of X
$\mid Y \mid$	ABS(Y)	Absolute value of Y
$[\![X]\!]$	INT(X)	Greatest integer less than or equal to X
=	=	Equals
≠	# or <>	Does not equal
<	<	Less than
>	>	Greater than
⩽	<=	Less than or equal to
⩾	>=	Greater than or equal to
←	=	Replaced by
( ) or [ ]	( ) or [ ]	Inclusive brackets or parentheses
A_i	A(I)	Subscripted variable
$A_{i,j}$	A(I,J)	Double subscripted variable
(None)	RND(X)	Assign a random number to the variable X

NUMBER SET DESIGNATIONS

N – Natural number set
W – Whole number set
I – Set of integers
Q – Rational number set
Z – Irrational number set
R – Real number set

DEFINITION AND PROPERTIES OF SYSTEMS

A mathematical system is a structure involving a non-empty set of elements on which:

(1) one or more binary operations have been defined, and

(2) a set of properties has been established which specifies certain relationships among elements of the set.

In this unit, we will study three mathematical systems. Each of the systems is based on a non-empty set of elements and two binary operations which we will call addition $\oplus$ and multiplication $\circledast$. We use the symbols $\oplus$ and $\circledast$ because the operations defined are not necessarily real number addition and multiplication. Each system will exhibit some or all of the following properties.

1. *Closure Property For Addition:* For x and y elements of S, there is a unique element of S (we'll call it z) such that $x \oplus y = z$.

2. *Closure Property For Multiplication:* For x and y elements of S, there is a unique element of S, z, such that $x \circledast y = z$.

3. *Associative Property For Addition:* For all x, y, z elements of S, $(x \oplus y) \oplus z = x \oplus (y \oplus z)$.

4. *Associative Property For Multiplication:* For all x, y, z elements of S, $(x \circledast y) \circledast z = x \circledast (y \circledast z)$.

5. *Identity Property For Addition:* For each x which is an element of S, there exists a unique element, i, such that $x \oplus i = x$.

6. *Identity Property For Multiplication:* For each x which is an element of s, there exists a unique element, j, such that $x \circledast j = x$.

7. *Commutative Property For Addition:* For all x, y elements of S, $x \oplus y = y \oplus x$.

8. *Commutative Property For multiplication:* For all x, y elements of S, $x \circledast y = y \circledast x$.

9. *Inverse Property For Addition:* For each x which is an element of S, there exists a unique element y such that $x \oplus y = i$ (i defined in (5) above).

10. *Inverse Property For Multiplication:* For each x which is an element of S, there exists a unique element y such that $x \circledast y = j$ (j defined in (6) above).

11. *Distributive Property:* For all x, y, z elements of S, $x \circledast (y \oplus z) = (x \circledast y) \oplus (x \circledast z)$.

SETS

Until recently, the study of sets was reserved for graduate students. Then it became apparent that the concept of sets illuminates many branches of mathematics. Today it is not unusual to find the concept and language of sets introduced at the elementary school level.

The algebra of sets was developed by the English mathematician George Boole (1815–1864), and is called Boolean Algebra. Over the years, mathematicians have found a wide range of applications of this algebra, particularly in the areas of logic and probability.

The mathematical system based on the set of sets has two operations defined, *union* ($\cup$) and *intersection* ($\cap$). Properties 1 through 11 (discussed in the previous section) are properties of this system.

The *union* of sets A and B yields a set C which includes all elements x such that x is an element of A or x is an element of B. In set notation, *union* is expressed as:

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$$

The *intersection* of sets A and B yields a set C which includes all elements x such that x is an element of both A and B. In set notation:

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$$

A is said to be a *subset* of B (expressed by $A \subseteq B$) if and only if every element, x, of A is also an element of B. In set notation:

$A \subseteq B$ if and only if $\forall x \in A, x \in B$. ($\forall x \in A$ is read for every x element of A)

We will now consider computer programs that illustrate the operations of union and intersection, as well as some of the properties of the system. Though most mathematical systems involve infinite sets, for obvious reasons our investigations will involve only finite examples. Even the computer cannot deal with an infinite number of elements. The conclusions we will draw from our finite examples also apply to infinite sets.

EXERCISE 1 – Determining Subsets

Write a computer program that will determine if A is a subset of B, where A and B are non-empty sets. Test your program on the following sets. Note that the null set is a subset of every set, and that every set is a subset of itself.

(1) $A = \{7, 16, .5, -3\}$; $B = \{.5, 7, -3, 13, 0, 16\}$

(2) $A = \{4, 2/5, 13\}$ $B = \{2/5, 4, 13\}$

(3) $A = \{-8, 3.1415, 10, 1, 5\}$ $B = \{-8, 3.1415, 10, 5\}$

(4) $A = \{17, -5, 2, .01\}$ $B = \{-7, .1, 3, 5\}$

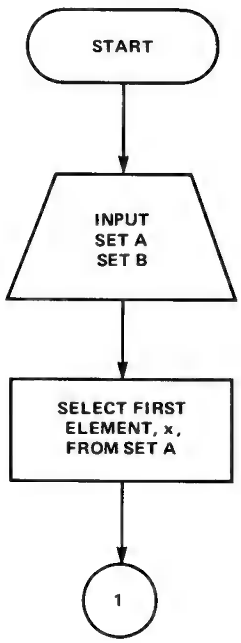
(5) $A = \phi$, $B = \{-7, 2\}$

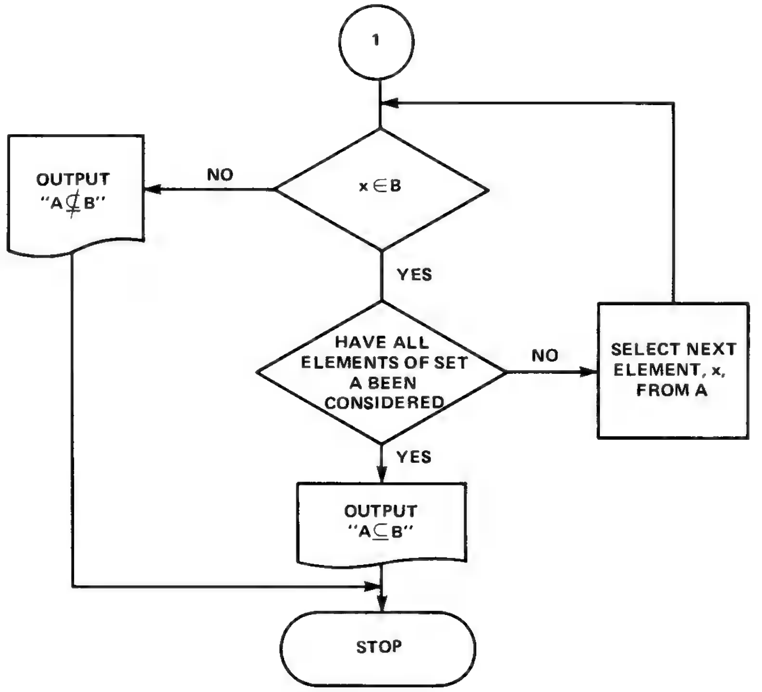
Problem Analysis

This exercise is solved simply by determining if each element of A is an element of B.

Macro Flow Chart

Exercise 1.





EXERCISE 2 – The Intersection of Sets

Write a computer program that will print the intersection of two non-empty sets A and B. Test your program on the following pairs of sets

(1) $A = \{3, -2.5, 7, .27\}$ $B = \{-.7, 3, -6, .5\}$

(2) $A = \{17, 3.14, -17\}$ $B = \{0, 17, -6, 3.14, -17\}$

(3) $A = \{7, -.66, 25, 6\}$ $B = \{-2, 0, 2.16\}$

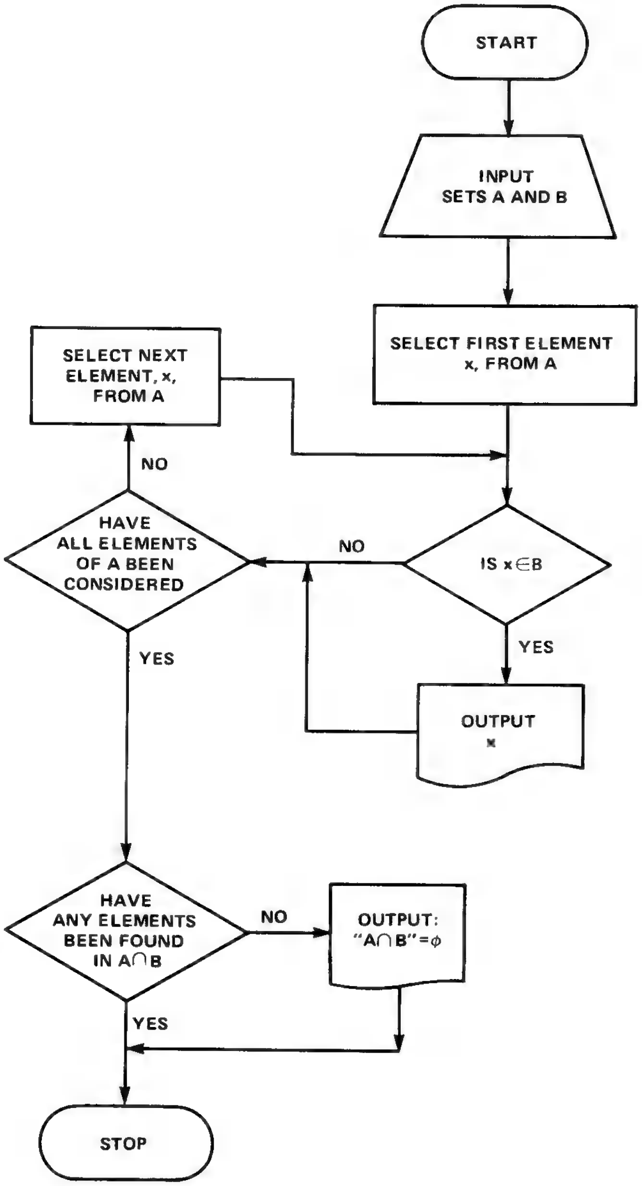
(4) $A = \{-5, 0, 1\}$; $B = \{0, -5, 1\}$

Problem Analysis

The procedure for this exercise is similar to that used in the previous exercise.

Macro Flow Chart

Exercise 2.



EXERCISE 3 – The Union of Sets

Write a computer program that will form the union of two non-empty sets A and B. Test your program on:

(1) $A = \{6, 7, -2, -1.6, 75\}$ $B = \{3, -18, -2, 75, 11\}$

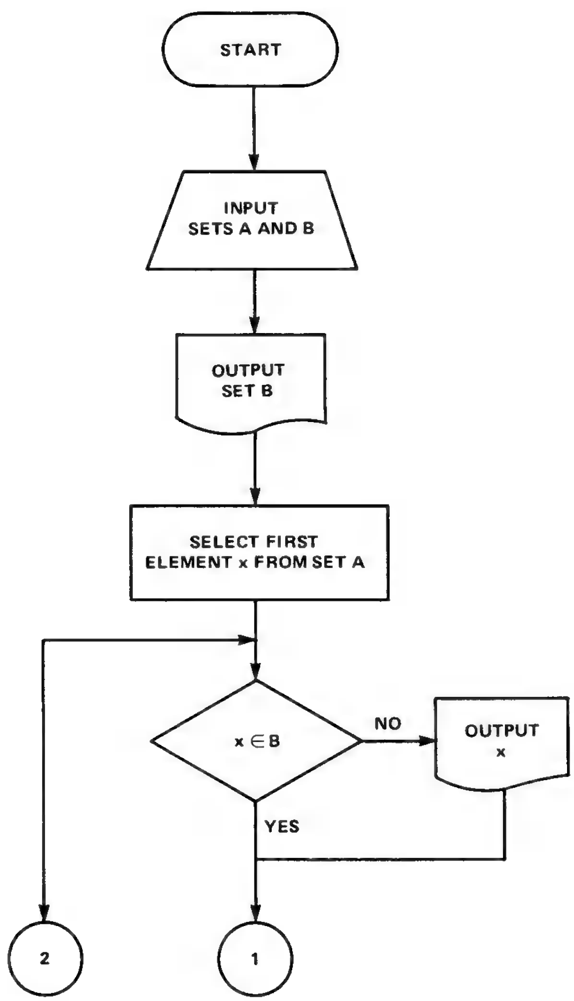
(2) $A = \{7, -7, 0, 1, -1\}$ $B = \{7, -7, 0, 1, -1\}$

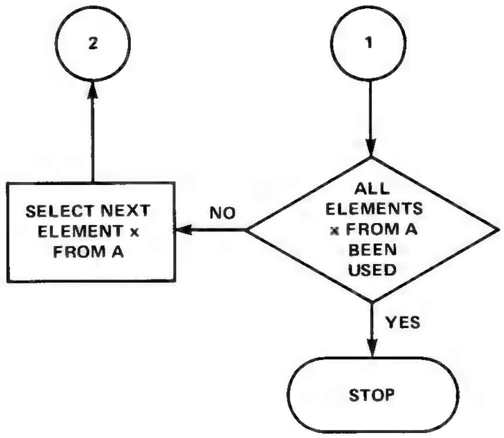
Problem Analysis

The flow chart clearly shows this simple procedure.

Macro Flow Chart

Exercise 3.





EXERCISE 4 – Modeling Commutativity of the Union Operation for Two Given Sets A and B.

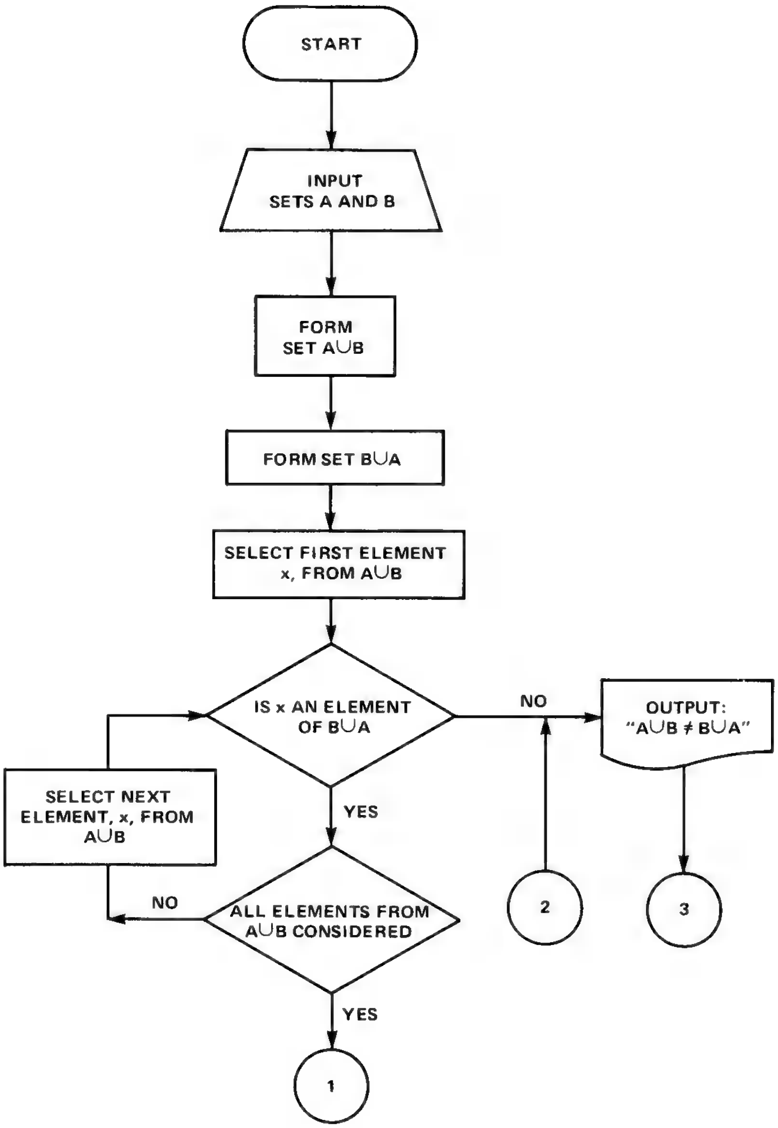
Write a program to determine if, for two non-empty finite sets A and B, $A \cup B = B \cup A$. Apply your program to the sets of Exercise 3.

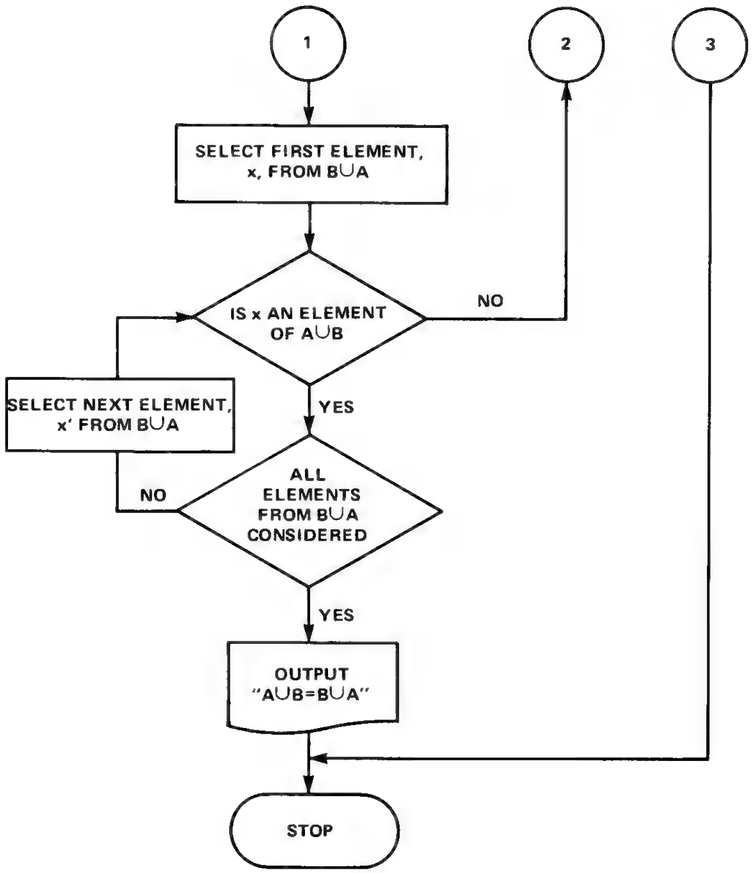
Problem Analysis

This problem involves testing each element of the two unions.

Macro Flow Chart

Exercise 4.





SUGGESTED REFERENCES FOR THIS SECTION

Allendoerfer, Carl B., and Oakley, Cletus O., *Principles of Mathematics*, 2nd Edition, McGraw-Hill, New York, 1963.

Courant, Richard, and Robbins, Herbert, *What Is Mathematics*, Oxford University Press, New York, 1958.

Dolciani, Mary P., et al., *Modern Introductory Analysis*, Houghton-Mifflin Co., Boston, 1964.

A Second Course in Algebra and Trigonometry with Computer Programming, Colorado Schools Computing Science Curriculum Development Project, 1969.

MATRICES

Like sets, matrices are a recent addition to high school mathematics. In fact, matrices have been a common tool of rank-and-file mathematicians and scientists for only a few years. The trend toward matrix study is due mostly to the development of the electronic computer. Applied mathematicians now make extensive use of matrices because electronic computers are able to perform matrix arithmetic which was impossible to perform with mechanical calculators.

The theory of matrices was originally derived by James J. Sylvester, an English mathematician of the mid-1800's. He is responsible for applying the term "matrix" to an array of terms or elements.

As matrix theory was expanded, mathematicians realized the value of matrices in extending the common notions of numbers. Applied mathematicians found matrices useful in their studies: for example, in 1925 a German physicist named Heisenberg introduced them into the field of quantum mechanics. Since the electronic computer was developed, the application of matrices has become so widespread that there is hardly a field of applied math where the concept is not used. This unit will concentrate on the fundamental operations and properties of matrices. Problems of application will be considered in other units of this series. We will be studying a system based on a set of square matrices. Our task is to determine which of the properties 1-11 apply to this system.

The table below defines symbols and terms applicable to our study of matrices.

Symbol or Term	Definition
Matrix	A rectangular array of elements, such as $\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}$
$A_{m,n}$	A matrix A, which has m rows and n columns. $A_{n,n}$ would be a square matrix. Capital letters without subscripts will be used when it is not necessary to indicate the number of rows and columns.
a_{ij}	The jth element in the ith row, i.e., the element at the intersection of row i and column j of matrix A.
$A_{m,n} + B_{m,n}$	This addition results in a matrix $C_{m,n}$ with elements $c_{ij} = a_{ij} + b_{ij}$ for i = 1 to m and j = 1 to n.
$A_{m,n} \times B_{n,m}$	This multiplication yields a matrix $C_{m,m}$ with elements $c_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + \ldots + a_{im} \times b_{mj}$ for i = 1 to m and j = 1 to n.
$A_{m,n} = B_{p,q}$	Matrices A and B are equal if and only if m = p, n = q, and $a_{ij} = b_{ij}$ for $1 \leqslant i \leqslant m$, and $1 \leqslant j \leqslant n$.
Transposition of Matrix A	This yields a matrix T such that $t_{ij} = a_{ji}$ for each element of A.
Zero Matrix	A matrix Z such that each term, z_{ij}, is equal to zero.

EXERCISE 5 – Modeling Matrix Definitions

This exercise is a series of problems which use the computer to model the basic definitions in the above table.

(a) Write a computer program to print the transposition of a matrix $A_{m,n}$. Test your program on the following matrix:

$$A = \begin{pmatrix} 1 & -7 & 0 & 8 \\ 3 & .5 & -3 & 7 \\ 0 & 1 & 15 & 23 \\ 8 & 2 & 5 & -.8 \end{pmatrix}$$

(b) Write a computer program which will determine if two given matrices A and B are equal. The program should provide for the condition that the matrices must be of the same order (have same number of rows and columns.) Test your program on the following pairs of matrices:

(1) $A = \begin{pmatrix} 3 & 2 & -1 \\ 6 & 8 & 0 \end{pmatrix}$ $B = \begin{pmatrix} 3 & 2 & -1 \\ 6 & 8 & 0 \\ -5 & 6 & .6 \end{pmatrix}$

(2) $A = \begin{pmatrix} 3 & 2 & -1 \\ 6 & 8 & 0 \\ -5 & 7 & .6 \end{pmatrix}$ $B = \begin{pmatrix} 3 & 2 & -1 \\ 6 & 8 & 0 \\ -5 & 6 & .6 \end{pmatrix}$

(3) $A = \begin{pmatrix} 7 & -2 & .5 & 1 \\ 6 & 0 & -7 & 0 \\ -.7 & 2 & 1 & 3 \\ 12 & 8 & 19 & -20 \end{pmatrix}$ $B = \begin{pmatrix} 7 & -2 & .5 & 1 \\ 6 & 0 & -7 & 0 \\ -.7 & 2 & 1 & 3 \\ 12 & 8 & .19 & -20 \end{pmatrix}$

(c) *Write a computer program that will add two matrices A and B. Apply your program to the following pairs of matrices.*

(1) $A = \begin{pmatrix} 8 & -2 & 5 \\ -20 & .5 & 1 \\ 0 & -7 & .8 \\ 6 & 1 & 0 \end{pmatrix}$ $B = \begin{pmatrix} 16 & 27 & 0 \\ -1 & 18 & 2 \\ 5 & -.16 & 3 \\ 0 & 1 & 7 \end{pmatrix}$

(2) $A = \begin{pmatrix} 3 & -7 & 1 & .8 \\ 2 & 7 & 0 & 5 \end{pmatrix}$ $B = \begin{pmatrix} 17 & 7 & 0 & .2 \\ 3 & 3 & 10 & 5 \end{pmatrix}$

(d) *Write a computer program that will multiply two matrices A and B. Apply your program to the following matrices:*

(1) $A = \begin{pmatrix} -3 & 0 & 1 \\ 2 & 1 & 3 \\ -2 & 0 & 1 \end{pmatrix}$ $B = \begin{pmatrix} 2 & 1 & -1 \\ 2 & 3 & 1 \\ -2 & 1 & 0 \end{pmatrix}$

(2) $A = \begin{pmatrix} 3 & -2 & 1 \\ 2 & 0 & -5 \end{pmatrix}$ $B = \begin{pmatrix} 2 & -5 \\ 0 & 1 \\ 2 & 2 \end{pmatrix}$

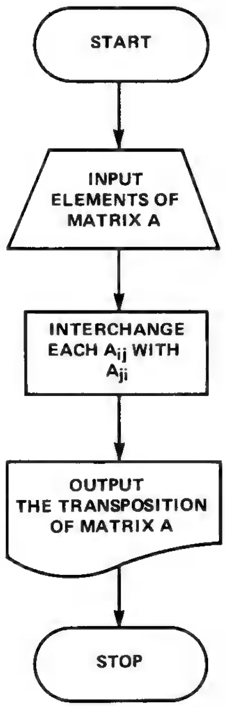
(3) $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$ $B = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$

Problem Analysis

You will need to understand the definition of each operation before you can successfully complete this exercise. The problems are straightforward applications of these definitions.

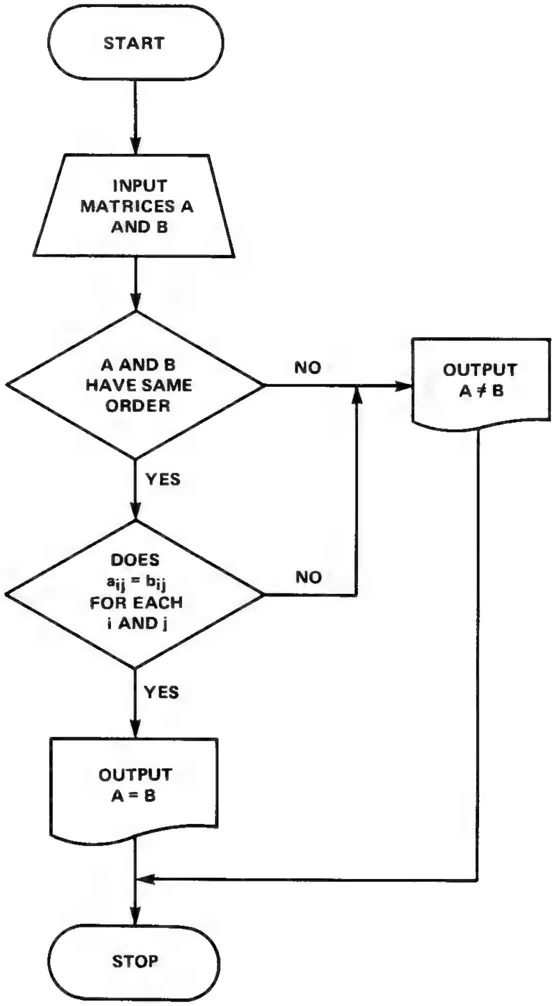
Macro Flow Chart

Exercise 5(a)

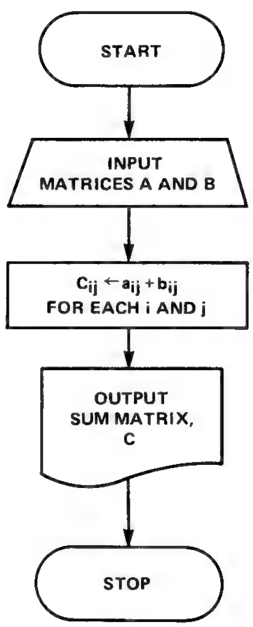


Macro Flow Chart

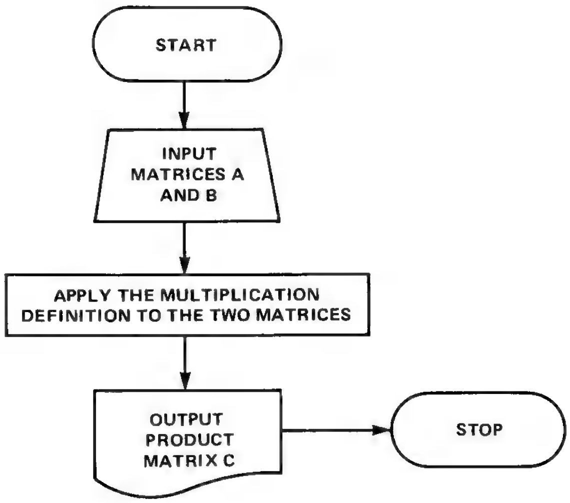
Exercise 5(b).



Exercise 5(c).



Exercise 5(d).



EXERCISE 6 – Properties of a System of Square Matrices

This exercise is a series of problems which demonstrate those algebraic properties that apply to the system of square matrices. It can be shown that nine out of the eleven properties do apply, while commutative and inverse properties for multiplication *do not. The exercise will deal with selected properties which do apply to the system. If you have time, write programs that demonstrate the other applicable properties. Remember that proving that a property applies to a finite number of cases does not prove that it applies to* all *cases.*

(a) *Write a computer program which will determine if $A + B = B + A$ for two $n \times n$ matrices. Use the definition of addition as given at the beginning of this section. Try your program on the following pairs of matrices:*

(1)
$$A = \begin{pmatrix} 8 & -2 & 5 \\ 20 & .5 & 1 \\ 0 & -7 & .8 \\ 6 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 16 & 27 & 0 \\ -1 & 18 & 2 \\ 5 & -.16 & 3 \\ 0 & 1 & 7 \end{pmatrix}$$

(2)
$$A = \begin{pmatrix} 3 & -7 & 1 & .8 \\ 2 & 7 & 0 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 17 & 7 & 0 & .2 \\ 3 & 3 & 10 & 5 \end{pmatrix}$$

(b) *This part of the exercise tests the commutative property for multiplication. Using the matrices given below, have the computer print out A × B and B × A. You will see that commutativity does not apply in at least one instance:*

(1)

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(2)

$$A = \begin{pmatrix} 6 & -2 & 1 \\ 0 & 1 & -1 \\ 3 & 2 & 1 \end{pmatrix} \quad B = \begin{pmatrix} -2 & 3 & 0 \\ 4 & 5 & 3 \\ -1 & 0 & 1 \end{pmatrix}$$

(c) *There is an identity matrix, I, for multiplication. It is the matrix in which each element a_{ii} equals one and all other elements equal zero. Example:*

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Write a program that will show that A × I = I × A = A. Test your program for:

(1)

$$A = \begin{pmatrix} -17 & 0 & 22 \\ 5 & -.83 & 1 \\ 6 & -1 & 0 \end{pmatrix}$$

(2)

$$A = \begin{pmatrix} 1 & -1 & 2 & -2 \\ 3 & -3 & 4 & -4 \\ 5 & -5 & 6 & -6 \\ 7 & -7 & 8 & -8 \end{pmatrix}$$

Although the commutative property does not hold for matrix multiplication in general, you will see from your program results that commutativity does apply when the multiplicative identity matrix is involved.

(d) *Write a program to determine if a given 2 × 2 matrix has a multiplication inverse. If it does, print the inverse matrix, expressing its elements in both radical and decimal form.*

Use your program to find the inverse of:

(1)

$$A = \begin{pmatrix} 17 & -3 \\ 2 & 1 \end{pmatrix}$$

(2)

$$A = \begin{pmatrix} 6 & -.5 \\ 36 & -3 \end{pmatrix}$$

Problem Analysis

Parts (a), (b), and (c) are obvious applications of the matrix definitions and the algebraic properties discussed earlier. Part (d) is more complicated and, consequently, we will discuss it at some length.

The multiplicative inverse property does not hold for matrices in general, but in practical applications when an inverse does exist it has a significant role. We will look at 2 X 2 matrices only for reasons of simplicity: determining if a higher order matrix has a multiplicative inverse is too involved to consider here. We will deal with this matter again in a later unit of this series.

For a matrix A to have a multiplicative inverse means that there exists a matrix A′ such that both A X A′ and A X A equal the identity matrix I. For example, consider the following matrix:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

where a, b, c, and d are real numbers. For A to have an inverse, we need to find the values x, y, u, w of a matrix A′, so that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} x & y \\ u & w \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Let's carry through the solution to this matrix.

$$\begin{pmatrix} ax + bu & ay + bw \\ cx + du & cy + dw \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ by definition of matrix multiplication.}$$

$$\left.\begin{array}{llll} (1) & ax + bu = 1 & (2) & cx + du = 0 \\ (3) & ay + bw = 0 & (4) & cy + dw = 1 \end{array}\right\} \text{ from definition of equality of matrices}$$

(2) $cx + du = 0 \rightarrow x -du/c$
(1) $a(-du/c) + bu = 1 \rightarrow u = -c/(ad - bc)$
(2) $cx + d\,(c/(bc - ad)) = 0 \rightarrow x = d/(ad - bc)$
(3) $ay + bw = 0 \rightarrow y = -\,(bw/a)$
(4) $c(-bw/a) + dw = 1 \rightarrow w = a/(ad - bc)$
(3) $ay + b\,(ad - bc)) = 0 \rightarrow y = -b/(ad - bc)$

We have determined that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} d/(ad - bc) & -b/(ad - bc) \\ -c/(ad - bc) & a/(ad - bc) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

i.e.,

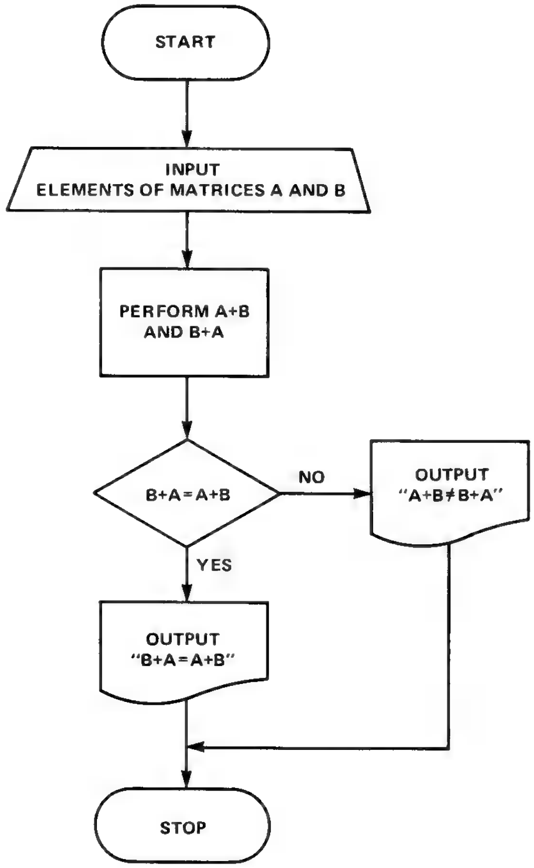
$$A^{-1} = \begin{pmatrix} d/(ad - bc) & -b/(ad - bc) \\ -c/(ad - bc) & a/(ad - bc) \end{pmatrix}.$$

Therefore A^{-1} is undefined when (ad – bc) equals zero. In this case, we say that a multiplicative inverse does not exist.

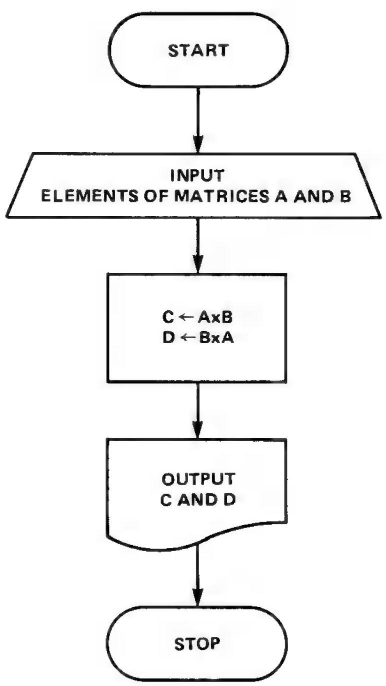
When you have found a valid inverse matrix, A^{-1}, check to see if $A^{-1} \times A = 1$. You will find the results rather interesting.

Macro Flow Chart

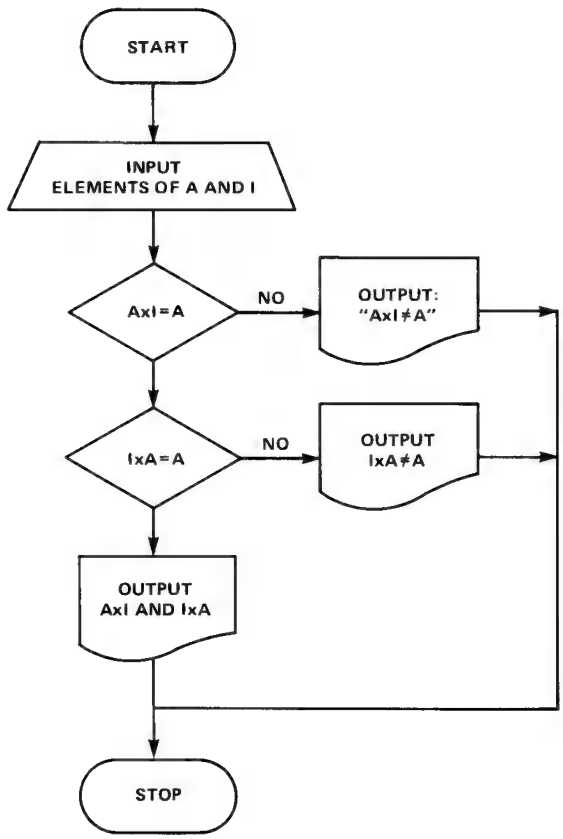
Exercise 6(a).



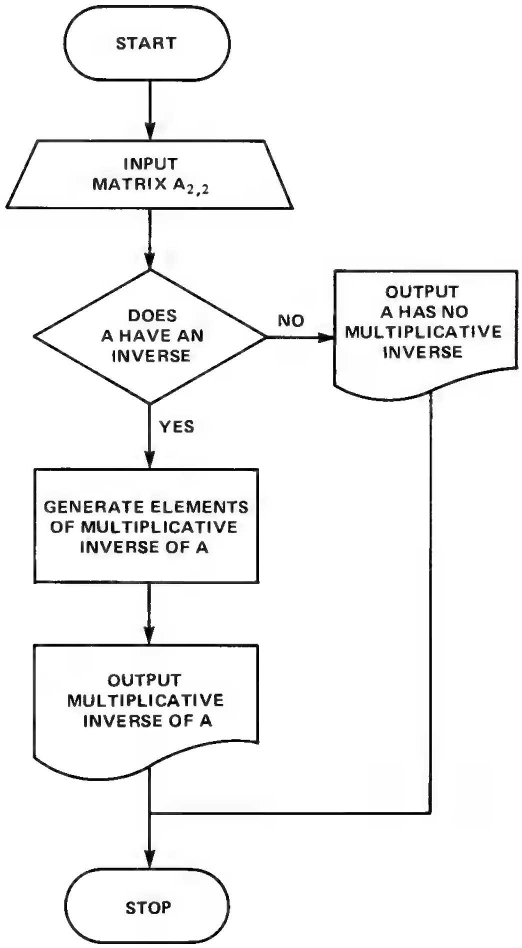
Exercise 6(b)



Exercise 6(c)


START
INPUT
ELEMENTS OF A AND I
AxI=A
NO
OUTPUT:
"AxI≠A"
IxA=A
NO
OUTPUT
IxA≠A
OUTPUT
AxI AND IxA
STOP

Exercise 6(d)



EXERCISE 7 – A Matrix Application

At the beginning of this section we indicated that matrices have many practical applications. One of these applications is to display data in a matrix form. We will look at a very simple example.

Suppose a contractor is being asked to lay a water line from point A to point B in a city. (See Figure 1 below.)

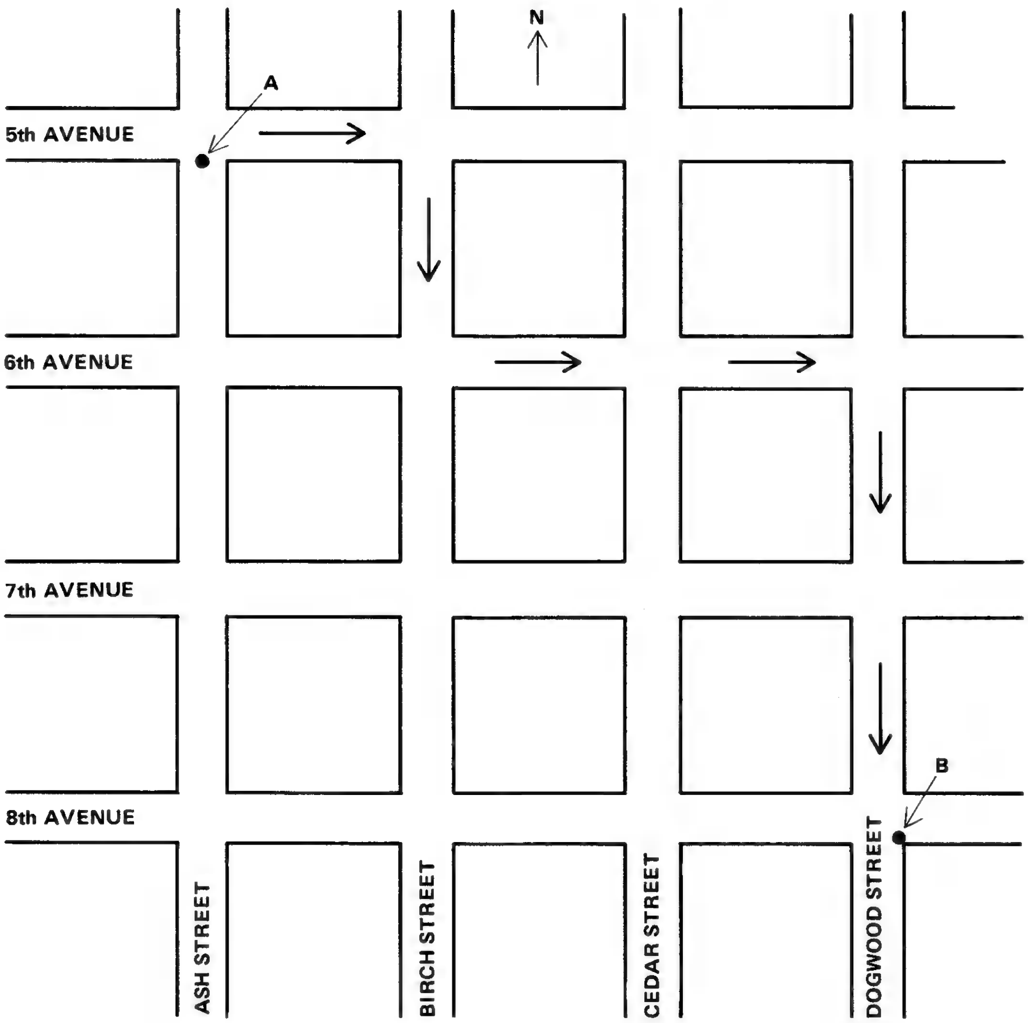


Figure 1. Street Layout of City

The restrictions placed on the contractor are that the ditch must be in the street and that the ditch must always be dug in an easterly and southerly direction. One possible path the contractor might follow is represented by the arrows in the figure.

(a) Write a computer program to determine how many different paths the water line can follow to go from point A to point B. Two paths are considered to be different if at any point the paths go in different directions. Write the program so it can be used for any number of horizontal and vertical streets.

(b) Run your program for 6 horizontal streets and 8 vertical streets.

Problem Analysis

(a) At each street intersection the contractor has the option of laying the line toward the east or toward the south. In Figure 2, the number recorded at each intersection is the number of paths that can be followed to go from point A to that intersection.

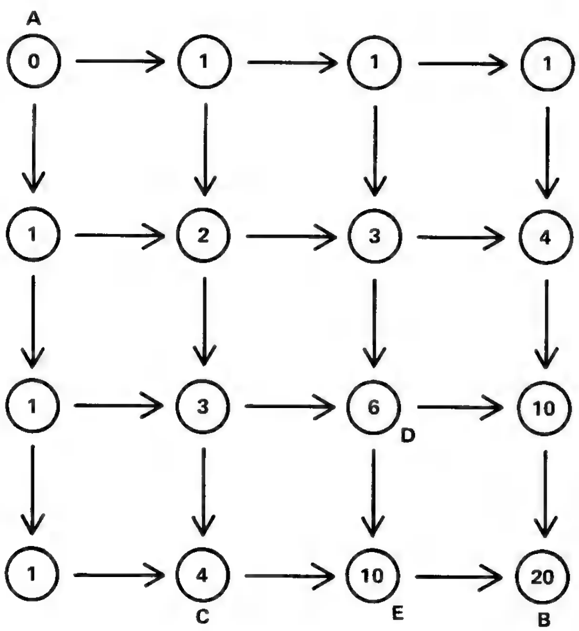


Figure 2. Symbolic Representation of the Contractor's Task

For instance, to go from A to E in Figure 2 we must pass through C or D. The figure indicates that we can get to point C in 4 different paths and to point D via 6 paths, from A. Therefore there are $4 + 6 = 10$ different paths to E.

One approach to the above problem is to begin with the matrix

$$P = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

called the constant matrix then transform it to

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{pmatrix}$$

We start the program by letting the first element p (1, 1) in matrix P equal 0. Leave the rest of row 1 and column 1 elements each equal to 1. Then

$$p(2,2) = p(1,2) + p(2,1) = 1 + 1 = 2$$

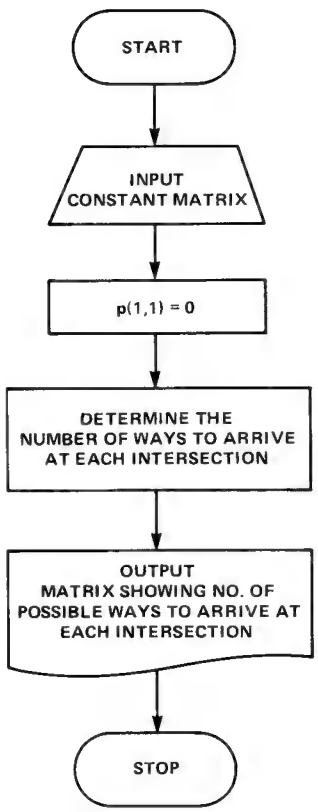
$$p(2,3) = p(1,3) + p(2,2) = 1 + 2 = 3$$

$$\vdots$$

$$p(j,k) = p(j-1,k) + p(j,k-1) \text{ for } j = 2 \text{ to } 4 \text{ and } k = 2 \text{ to } 4$$

Macro Flow Chart.

Exercise 7.



SUGGESTED REFERENCES FOR THIS SECTION

Allendoerfer, Carl B., and Oakley, Cletus O., *Principles of Mathematics,* 2nd Edition, McGraw-Hill, New York, 1963.

A Second Course in Algebra and Trigonometry with Computer Programming, Colorado Schools Computing Science Curriculum Development Project, 1969.

MODULAR ARITHMETIC

Modular arithmetic, sometimes called "clock" arithmetic, is a mathematical system defined for an arbitrary, but finite, set of numbers. Since the number of elements within each modulus is finite, we can use the computer to verify the properties which apply to the system.

To illustrate modular arithmetic, we choose some natural number, say 4, as the modulus. In this case, the numbers 0, 1, 2, 3 are elements of the system. We will use a 4-hour clock to define addition for our system.

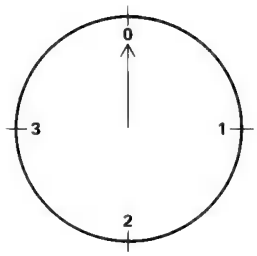


Figure 3. A 4-Hour Clock Used to Illustrate Modular Arithmetic

To add 3 + 2 we set the hand to 3, then move it 2 spaces clockwise and the hand will then rest on 1. We will then say the sum of 2 + 3 = 1 in our Mod (4) system. The same result could be achieved by performing real number addition on 3 + 2 to get 5, then reduce 5 by the modulus 4 to arrive at the sum 1. We can not use 5 as the sum because then the system would not be closed for addition. If the real sum of two elements is less than 4 then no reduction would be necessary and the real number sum would become the Mod (4) sum.

Example: 3 + 0 = 3 in real number addition. 3 + 0 also equals 3 in Mod (4) addition.

For multiplication, all real products greater than or equal to 4 for any two elements of the system are reduced by subtracting some multiple of 4 so that the remainder is less than or equal to 3.

The addition and multiplication tables for Mod (4) arithmetic are given below.

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

x	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

We can see from the tables that Mod (4) arithmetic is a closed system, i.e., there is closure for addition and multiplication. But the other algebraic properties are not so readily verified. We will examine some of these other properties via the computer in the following exercises.

EXERCISE 8 – Commutativity of Mod (4) Arithmetic

(a) Write a computer program that will determine if Mod (4) addition is commutative.

(b) Using the approach suggested in the Problem Analysis, change the value of element a_{34} to 3. Run your program with this change. The system is no longer commutative.

(c) Adjust your program so the non-commutative cases are displayed.

(d) What adjustments would need to be made in your program to determine if multiplication is commutative in Mod (4) arithmetic?

Problem Analysis

(a) Examine the addition table

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Your task is to test if a + b = b + a for each a and b that are elements of Mod (4). That is, you must determine if according to the above table

0+0=0+0	1+0=0+1	3+0=0+3
0+1=1+0	1+1=1+1	3+1=1+3
0+2=2+0	:	
0+3=3+0	1+3=3+1	3+3=3+3

We see that there are 4 possible choices for a and 4 for b. Therefore a total of 4 X 4 or 16 cases would need to be verified to establish that Mod (4) addition is commutative. We can easily verify commutativity for a Mod (4) system, but the task becomes burdensome as the number of elements in the system increases.

We will use matrices to verify commutativity for the above system.

The interior of the above addition table is a 4 X 4 matrix.

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{pmatrix}$$

Assigning variables to the elements of the above matrix, we have

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

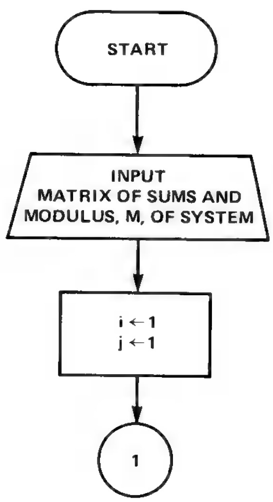
Let us examine what happens when we test some instances of addition for commutativity.

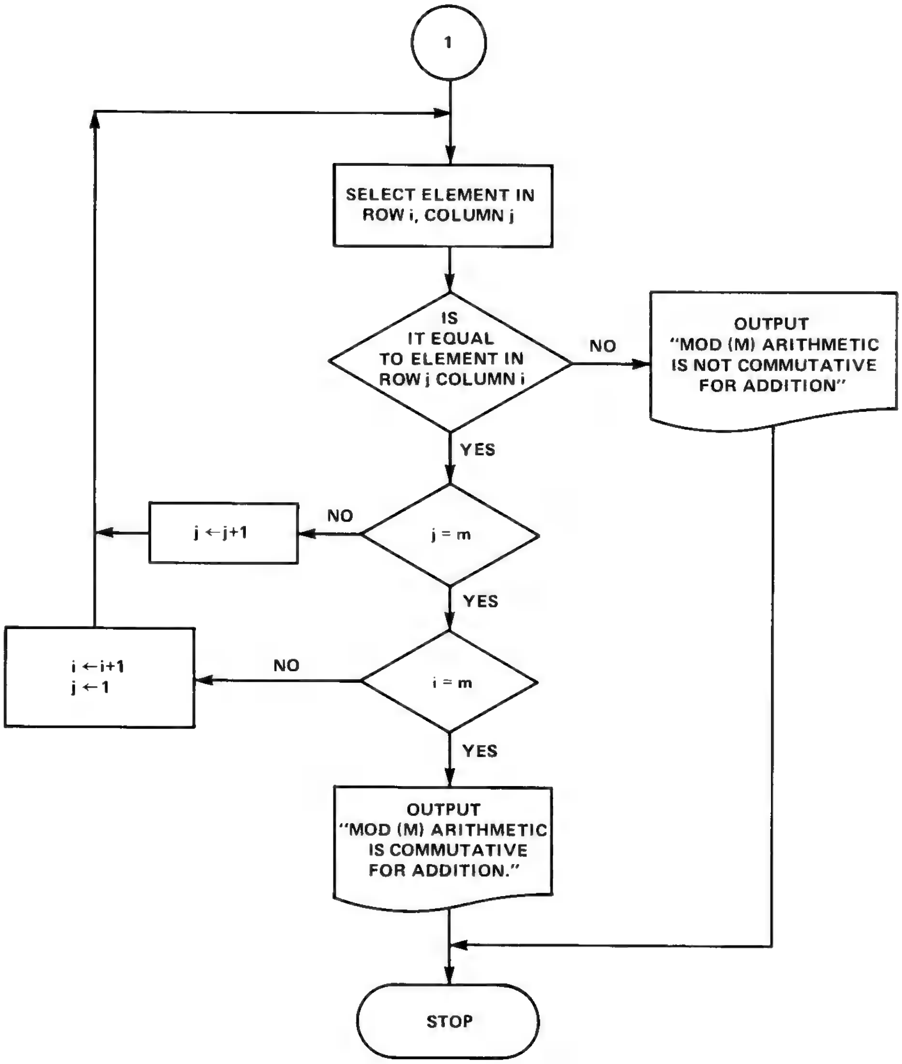
From the table we see that 2 + 3 = the number that has been assigned variable a_{34}. The number that has been assigned variable a_{43} = 3 + 2. Therefore, 2 + 3 = 3 + 2 if and only if $a_{34} = a_{43}$. Similarly, 1 + 2 = the number assigned the variable a_{23} and 2 + 1 = the number assigned the variable a_{32}. Therefore, 1 + 2 = 2 + 1 if and only if $a_{23} = a_{32}$.

From these instances, we can deduce that addition is commutative for a given system if $a_{ij} = a_{ji}$ for each element in the matrix of sums. Your program should input the matrix of sums and check each element.

Macro Flow Chart

Exercise 8.





EXERCISE 9 – Associativity of Mod (4) and (5) Arithmetic

(a) Write a computer program that will model the associative property for multiplication in Mod (M) arithmetic. Apply your program to Mod (4) and Mod (5) multiplication

(b) In the Mod (5) system change the a_{34} element to have the value 3. Run your program again. You should now get some non-associative cases.

Problem Analysis

Examine the Mod (4) multiplication matrix and the corresponding matrix of assignments.

x	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

x	a_1	a_2	a_3	a_4
a_1	a_{11}	a_{12}	a_{13}	a_{14}
a_2	a_{21}	a_{22}	a_{23}	a_{24}
a_3	a_{31}	a_{32}	a_{33}	a_{34}
a_4	a_{41}	a_{42}	a_{43}	a_{44}

The system is associative if for each i, j, and k

$$(a_i * a_j) * a_k = a_i * (a_j * a_k)$$

We must determine if:

(1) $(0*0) * 0 = 0 * (0*0) \leftrightarrow (a_1 * a_1) * a_1 = a_1 * (a_1 * a_1)$

$$0 * 0 = 0 * 0 \leftrightarrow \begin{cases} a_{11} * a_1 = a_1 * a_{11} \\ a_1 * a_1 = a_1 * a_1 \end{cases}$$

$$0 = 0 \leftrightarrow \begin{cases} a_{11} = a_{11} \\ a_1 = a_1 \end{cases}$$

(2) $(0*0) * 1 = 0 * (0*1) \leftrightarrow (a_1 * a_1) * a_2 = a_1 * (a_1 * a_2)$

$$0 * 1 = 0 * 0 \leftrightarrow \begin{cases} a_{11} * a_2 = a_1 * (a_{12}) \\ a_1 * a_2 = a_1 * a_1 \end{cases}$$

$$0 = 0 \leftrightarrow \begin{cases} a_{12} = a_{11} \\ a_1 = a_1 \end{cases}$$

(3) $(1*3) * 2 = 1 * (3*2) \leftrightarrow (a_2 * a_4) * a_3 = a_2 * (a_4 * a_3)$

$$3 * 2 = 1 * (2) \leftrightarrow \begin{cases} a_{24} * a_3 = a_2 * a_{43} \\ a_4 * a_3 = a_2 * a_{43} \end{cases}$$

$$2 = 2 \leftrightarrow \begin{cases} a_{43} = a_{23} \\ a_3 = a_3 \end{cases}$$

We abstract from the above cases the following algorithm:

If $(a_i * a_j) * a_k = a_i * (a_j * a_k)$

then $a_{ij} * a_k = a_i * (a_{jk})$.

But if the Mod (M) system is closed

$$a_{ij} = a_x \text{ for some x, } 0 \leqslant x \leqslant M - 1$$

$$a_{jk} = a_y \text{ for some y, } 0 \leqslant y \leqslant M - 1 ,$$

Then $$a_x * a_k = a_i * a_y$$

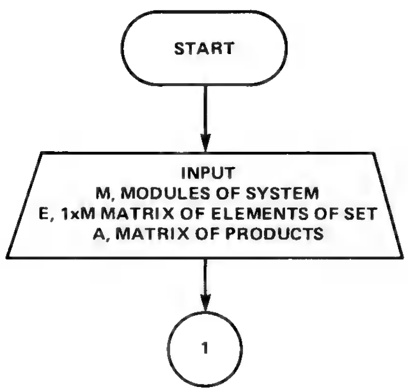
and $$a_{xk} = a_{iy} .$$

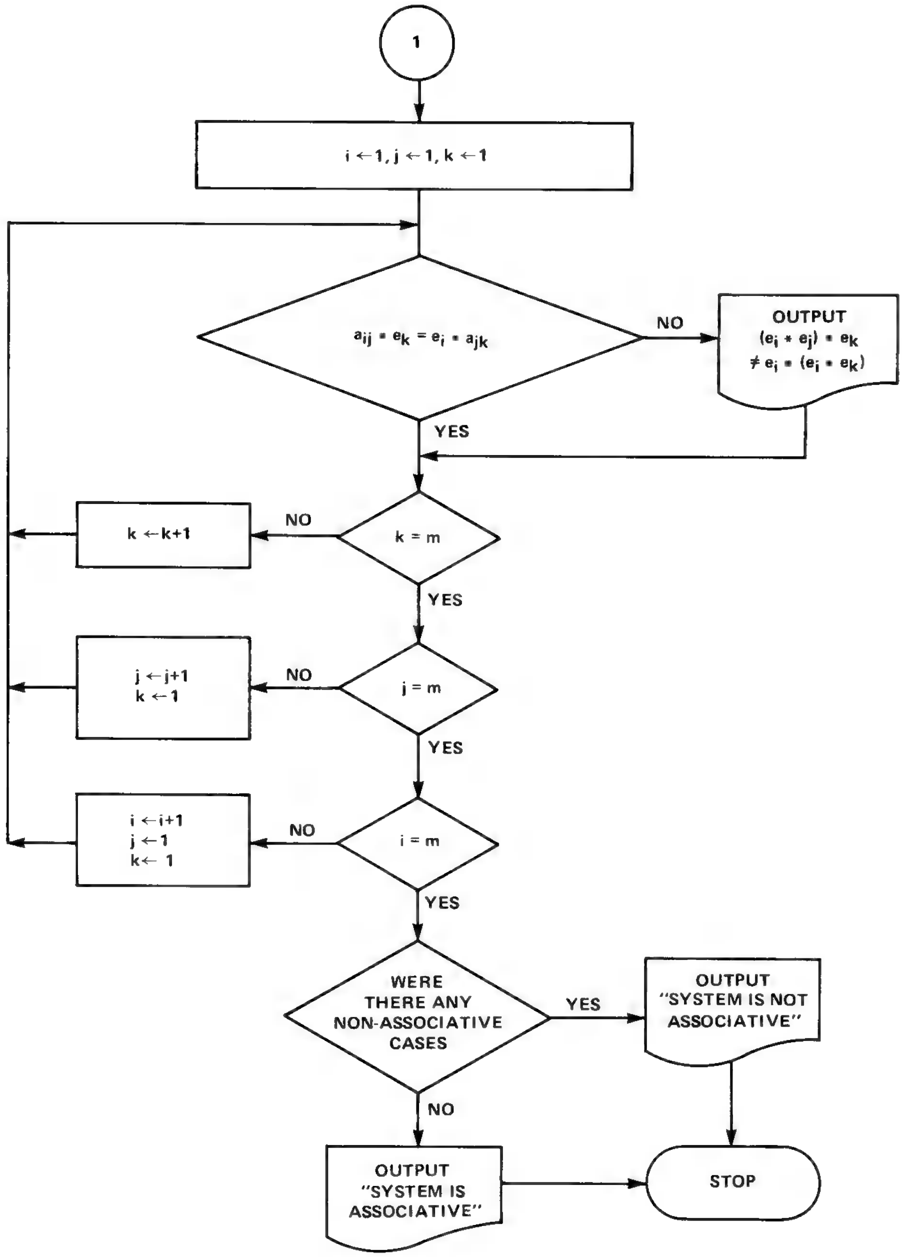
Therefore $\quad (a_i * a_j) * a_k = a_i * (a_j * a_k)$

Let's look at a flow chart of the above process.

Macro Flow Chart

Exercise 9.





EXERCISE 10 – Multiplicative Inverse for Mod (M) Arithmetic

(a) Write a computer program that will form the product matrix for Mod (M) arithmetic. Apply your program to producing the product matrices for Mod (4) and Mod (5).

(b) Add a sub-program to the above so that the computer will determine from the product matrix if the Mod (M) system has the multiplicative inverse property for a given modulus. Run the program for M=2,3,4,5,6, 7,8,9. Based on the output of the 8 runs, what do you think about the multiplicative inverse property for modular arithmetic?

(c) Do any of the modular arithmetic systems have all eleven properties?

Problem Analysis

(a) Your task is to have the computer find the product matrix P where

$$P = \begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ M-1 \end{pmatrix} \times (0, 1, 2, \ldots M-1) .$$

Then reduce each element of P by modules of M to arrive at the product matrix of Mod (M) multiplication.

The greatest integer function can be useful when reducing the real number products to Mod (M) products.

Let

$$\begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ M-1 \end{pmatrix} = \text{Matrix E}$$

and $(0, 1, 2, \ldots M-1) =$ Matrix F.

(b) The multiplicative inverse property states that for $a \neq$ the additive identity element, there exists a unique a' such that $a' = I = a'a$, I the identity element.

Assume we were examining the Mod (4) system in relation to the multiplicative inverse property. The Mod (4) multiplication table could be represented by

X	a_1	a_2	a_3	a_4
a_1	a_{11}	a_{12}	a_{13}	a_{14}
a_2	a_{21}	a_{22}	a_{23}	a_{24}
a_3	a_{31}	a_{32}	a_{33}	a_{34}
a_4	a_{41}	a_{42}	a_{43}	a_{44}

where Matrix E is represented by

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}$$

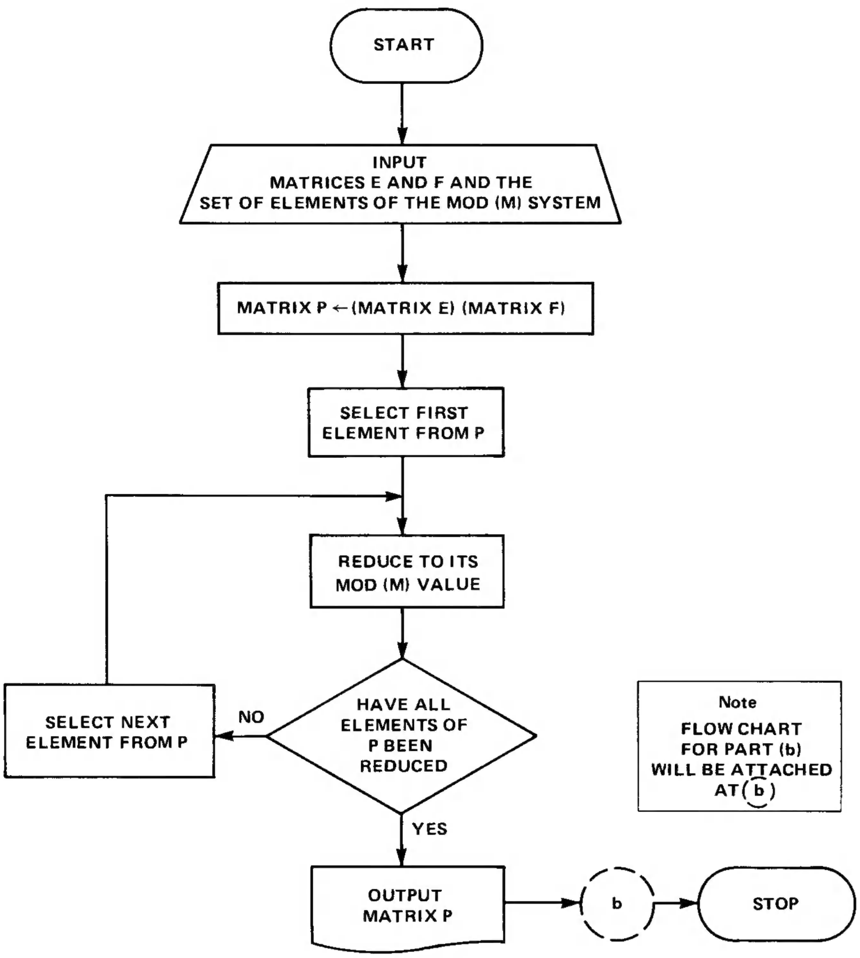
matrix F by (a_1, a_2, a_3, a_4), and the product matrix P by:

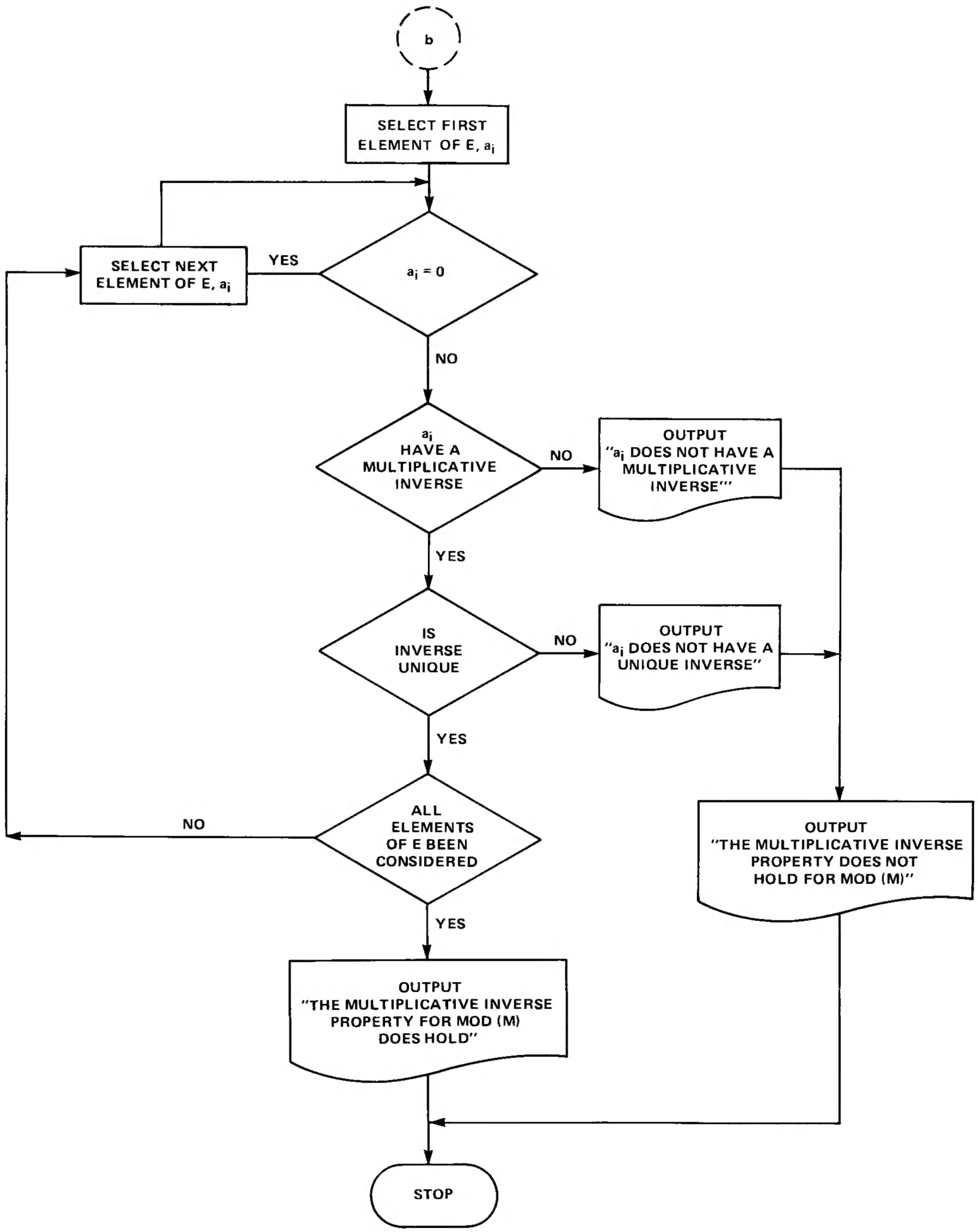
$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44}\,. \end{pmatrix}$$

Your program needs to search the matrix P to find if there is only one element per row which is equal to I. The additive identity element of the system is excluded from the definition of the multiplicative inverse. If there is no $a_{ij} = I$ in the row opposite a_i, there is no multiplicative inverse for a_i and the multiplicative inverse property does not hold for the system. If $a_{ij} = I$ for some a_i, then a_j is the multiplicative inverse. If more than one element in a row equals I, the multiplicative inverse property does not hold for the system.

Macro Flow Chart

Exercise 10 (a)





SUGGESTED REFERENCES FOR THIS SECTION

Allendoerfer, Carl B., and Oakley, Cletus O., *Principles of Mathematics*, 2nd Edition, McGraw-Hill, New York, 1963.

Davis, Philip J., *The Mathematics of Matrices*, 2nd Edition, Ginn and Co., New York, 1965.

Dolciani, Mary P., *et al., Modern Introductory Analysis*, Houghton-Mifflin Co., Boston, 1964.

Kemeny, John G., and Snell, J. Laurie, *Introduction to Finite Mathematics,* Prentice-Hall Inc., New York, 1957.

NOTES

NOTES